

## CLAIMS

1. In a system-level design and integration system which facilitates composing behavior, capturing architecture and mapping behavior onto architecture, a method for specification of scheduling for simulation and implementation of consumer embedded systems; the method comprising the steps of:

- a) identifying the schedulers in an architecture;
- b) identifying the schedulables constituting behavior;
- c) assigning schedulables and schedulers to each other;
- d) arranging for the schedulers to find their respective assigned schedulables;
- e) arranging for the schedulables to find their respective assigned scheduler;
- f) sending an event to a schedulable;
- g) sending an activation notice from the schedulable in step f) to its assigned scheduler;
- h) sending a message from the assigned scheduler of step g) to the schedulable of step f) to start its behavior's reaction; and
- i) sending a finish notice from the schedulable of step f) to the scheduler of step g) when said reaction is completed.

2. The method recited in claim 1 further comprising the steps of:

h1) having the scheduler of step g) determine whether the reaction of the schedulable of step f) must be preempted before it is finished;

h2) if in step h1) preemption is required, sending a suspend message from the scheduler of step g) to the schedulable of step f) to temporarily halt the execution of its reaction;

h3) if in step h1) preemption is required, causing the schedulable of step f) to halt its reaction in response to the suspend message;

h4) if in step H1) preemption is required, eventually sending a resume message from the scheduler of step g) to the schedulable of step f).

3. The method recited in claim 2 further comprising the steps of:

providing a next function which occurs when a scheduler receives an activation notice from an assigned schedulable while said scheduler is idle and when an assigned schedulable has finished a reaction;

said next function sending a handle to a schedulable which will be starting its behavior's reaction next.

4. The method recited in claim 2 further comprising the steps of:

providing a "preempt" function which occurs when an activation notice is received by a scheduler while its assigned schedulable is already reacting;

said "preempt" function determining whether a preemption should occur and whether such a preemption should be accomplished by suspending or aborting the reaction to be preempted.

5. The method recited in claim 2 further comprising the steps of:

providing a "next" function which occurs when a scheduler receives an activation notice from an assigned schedulable while said scheduler is idle and when an assigned schedulable has finished a reaction;

said next function sending a handle to a schedulable which will be starting its behavior's reaction next;

providing a preempt function which occurs when an activation notice is received by a scheduler while its assigned schedulable is already reacting;

said preempt function determining whether a preemption should occur and whether such a preemption should be accomplished by suspending or aborting the reaction to be preempted.

6. The method recited in claim 5 further comprising the steps of:

assigning transition times for completing selected states in said scheduler and schedulable operations including the steps of starting and resuming reactions, finishing reactions, and suspending and aborting reactions.

7. The method recited in claim 1 further comprising the step of:  
scheduling behaviors of said schedulables based in part on the priorities of mapping assignments of said schedulables.

8. The method recited in claim 2 wherein steps h1) to h4) are carried out in accordance with a cyclo-static scheduling policy.

9. The method recited in claim 2 wherein steps h1) to h4) are carried out in accordance with a non-preemptive static priority scheduling policy.

10. The method recited in claim 2 wherein steps h1) to h4) are carried out in accordance with a preemptive static priority scheduling policy.

11. The method recited in claim 2 wherein steps h1) to h4) are carried out in accordance with a selected scheduling policy.

12. The method recited in claim 11 further comprising the step of altering said selected scheduling policy based upon behavior of said embedded systems.

13. The method recited in claim 11 further comprising the step of generating software code for a consumer device to implement said selected scheduling policy.

14. The method recited in claim 11 further comprising the step of implementing said scheduling policy by interfacing to existing commercially available software.

15. A method for generating scheduling implementation for an embedded system expressed as an architecture, a behavior and a mapping of the behavior into the architecture; the method comprising the steps of:

- a) identifying the schedulers in an architecture;
- b) identifying the schedulables constituting behavior;
- c) assigning schedulables and schedulers to each other;
- d) arranging for the schedulers to find their respective assigned schedulables;
- e) arranging for the schedulables to find their respective assigned scheduler;
- f) sending an event to a schedulable;

g) sending an activation notice from the schedulable in step f) to its assigned scheduler;

h) sending a message from the assigned scheduler of step g) to the schedulable of step f) to start its behavior's reaction; and

i) sending a finish notice from the schedulable of step f) to the scheduler of step g) when said reaction is completed.

16. The method recited in claim 15 further comprising the steps of:

h1) having the scheduler of step g) determine whether the reaction of the schedulable of step f) must be preempted before it is finished;

h2) if in step h1) preemption is required, sending a suspend message from the scheduler of step g) to the schedulable of step f) to temporarily halt the execution of its reaction;

h3) if in step h1) preemption is required, causing the schedulable of step f) to halt its reaction in response to the suspend message;

h4) if in step H1) preemption is required, eventually sending a resume message from the scheduler of step g) to the schedulable of step f).

17. The method recited in claim 16 further comprising the steps of:

providing a next function which occurs when a scheduler receives an activation notice from an assigned schedulable while said scheduler is idle and when an assigned schedulable has finished a reaction;

said next function sending a handle to a schedulable which will be starting its behavior's reaction next.

18. The method recited in claim 16 further comprising the steps of:

providing a preempt function which occurs when an activation notice is received by a scheduler while its assigned schedulable is already reacting;

said preempt function determining whether a preemption should occur and whether such a preemption should be accomplished by suspending or aborting the reaction to be preempted.

19. The method recited in claim 16 further comprising the steps of:

providing a next function which occurs when a scheduler receives an activation notice from an assigned schedulable while said scheduler is idle and when an assigned schedulable has finished a reaction;

said next function sending a handle to a schedulable which will be starting its behavior's reaction next.

providing a preempt function which occurs when an activation notice is received by a scheduler while its assigned schedulable is already reacting;

said preempt function determining whether a preemption should occur and whether such a preemption should be accomplished by suspending or aborting the reaction to be preempted.

20. The method recited in claim 19 further comprising the steps of:

assigning transition times for completing selected states in said scheduler and schedulable operations, including the steps of starting and resuming reactions, finishing reactions, and suspending and aborting reactions.

21. The method recited in claim 15 further comprising the step of:

scheduling behaviors of said schedulables based in part on the priorities of mapping assignments of said schedulables.

22. The method recited in claim 16 wherein steps h1) to h4) are carried out in accordance with a cyclo-static scheduling policy.

23. The method recited in claim 16 wherein steps h1) to h4) are carried out in accordance with a cyclo-static scheduling policy.



24. The method recited in claim 16 wherein steps h1) to h4) are carried out in accordance with a preemptive static priority scheduling policy.

25. The method recited in claim 16 wherein steps h1) to h4) are carried out in accordance with a selected scheduling policy.

26. The method recited in claim 25 further comprising the step of altering said selected scheduling policy based upon behavior of said embedded systems.

27. The method recited in claim 25 further comprising the step of generating software code for a consumer device to implement said selected scheduling policy.

28. The method recited in claim 25 further comprising the step of implementing said scheduling policy by interfacing to existing commercially available software.